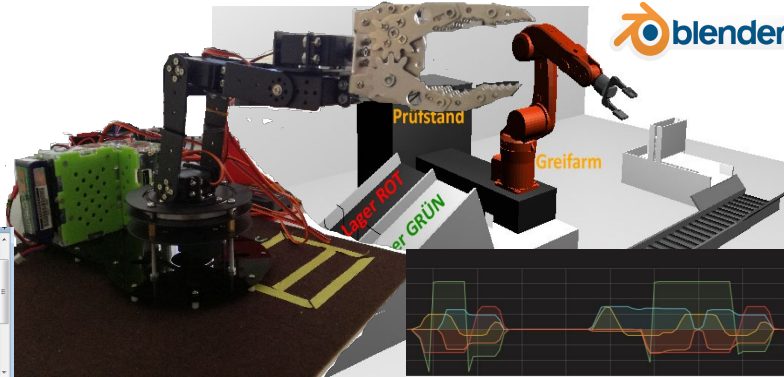
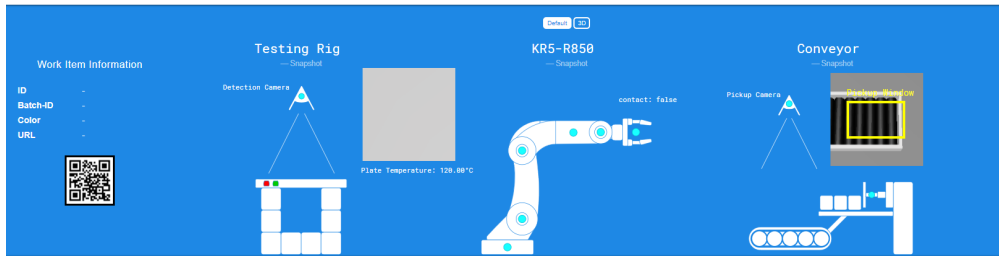


Prototypes



Saved Jobs

Pick, Heat & ...

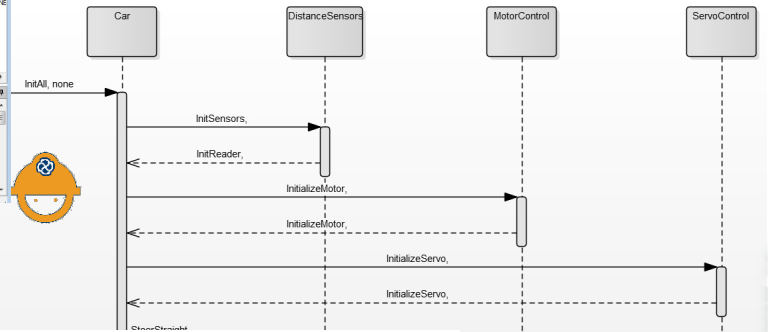
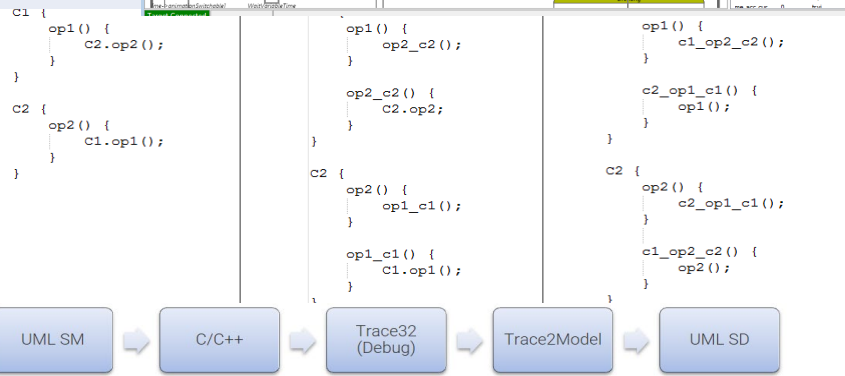
```

+ counter: int = 0
+ step: int = 5
+ to: int = 50
+ fmuStepSize: double = 1

+ inc(): int
+ constructor
+ init(int): void
    
```

```

215:         break;
217:     }
218:     break;
219:     case Millionrad_MainLogic_Standby:
220:     {
221:
222:     case Millionrad_MainLogic_DecidedNextIdleAnimation:
223:     /* DecideNextIdleAnimation -> EntryPoint */
224:     step->Standby.activeSubstate = Millionrad_MainLogic_IdleAnimations;
225:     step->IdleAnimations.startTime = FSM_getTime();
226:     if((me->animationCycle % 5) == 0)
227:     {
    
```



caseID	timestamp	lifeline	activity	messageParam	reqOrRes
1	2017-02-27 17:38:13.914	Car	InitAll	none	REQ
1	2017-02-27 17:38:13.917	DistSensors	InitSensors	none	REQ
1	2017-02-27 17:38:13.928	DistSensors	InitReader	none	RES
1	2017-02-27 17:38:13.954	ServoCont	InitServo	none	REQ
1	2017-02-27 17:38:13.964	ServoCont	InitServo	none	RES
1	2017-02-27 17:38:13.991	Car	SteerStraight	none	REQ
1	2017-02-27 17:38:13.992	ServoCont	SteerTo	dir=7	REQ
1	2017-02-27 17:38:15.145	ServoCont	SteerTo	dir=7	RES

- TimeSeriesCollectorRoleClassLib
- TraceLogEntryConfiguration { Class Resource }
- TimeSeriesCollectorInterfaceClassLib
- TracedValues { Class }

```

context Measurement::transportationCycleTime:
    →Float
Measurement.allInstances() -> sortBy(t) ->
select(m1,m2 | m1.isActive == 1 and
m2.isActive == 0 and m2.t > m1.t) ->
collect(m1, m2 | m2.t-m1.t) -> median()
    
```

